ISSN: 2688-8645

Automatic Quality Assessment of AIGC 3D Game Assets for Production Readiness and Its In-Engine Implementation

Danyang Wang [1,*], Xianyuan Liu [2]

[1] School of Digital Media, Guangzhou University of Software, Guangzhou, Guangdong, 510990, China

[2] College of Electronic Information and Control Engineering, Guangzhou University of Software, Guangzhou, Guangdong, 510990, China

* Corresponding author: Danyang Wang

Abstract

AIGC-based 3D generators have lowered the barrier of game asset production, yet their outputs frequently contain production-blocking defects such as non-manifold edges, flipped normals and overlapping UVs, while studios still rely on manual visual inspection. This paper presents a production-oriented quality assessment method for 3D game assets together with an in-engine implementation. A balanced 120-sample validation set with four severity levels is constructed by gradient defect injection on six kinds of procedural primitives. Twenty-nine geometric descriptors are extracted to train Random Forest and SVM-RBF classifiers, which are compared with a pure rule-based baseline. A dual-channel architecture that combines interpretable geometric rules with multi-view semantic scoring is designed, and the semantic channel is calibrated with real CLIP ViT-B/32 weights on CPU. The whole pipeline is encapsulated as a UE5 Python editor plugin. To prevent variants of the same prototype from crossing folds, repeated five-fold cross-validation grouped by prototype is adopted. Random Forest reaches 88.00%±0.70% accuracy and 0.8792±0.0071 F1 on the usable/unusable task, against 66.67% for the rule baseline; the grouped protocol is 2.67 percentage points higher than the stratified one, so no leakage-driven inflation is observed, and McNemar's test finds no significant difference from SVM-RBF (85.83%, p=0.581). The proportion of samples judged usable decreases monotonically across the four severity levels (100%, 73.3%, 20.0% and 0%). Winding inconsistency ratio (0.2452), inward-face ratio (0.1273) and open-edge ratio (0.1271) are the most informative descriptors. The semantic channel shows two-sided behaviour: text-only zero-shot scoring is not discriminative (AUC 0.27, with scores rising as defects worsen), whereas the multi-view embedding distance to clean reference renders reaches AUC 0.899 and 84.2% accuracy, approaching the geometric channel (AUC 0.919). Fusing both channels raises AUC only to 0.9228, indicating that they are largely redundant rather than complementary. CLIP encoding costs 73.31 ms per image (224x224, CPU, batch 8). In a real UE5 5.4.4 project, the plugin enumerates 1873 assets, loads 223 static meshes, and completes the export-inference-metadata-writeback-verification loop for 8 samples with an 8/8 success rate in 6.52 s. All reported separability is confined to the synthetic injection protocol and must be re-examined on real generated assets.

Keywords

AIGC; 3D Game Asset; Quality Assessment; Defect Injection; Geometric Descriptors; Unreal Engine.

ISSN: 2688-8645

1. Introduction

Text- or image-conditioned 3D generation has matured rapidly. DreamFusion[1], Shap-E[2] and commercial services such as Tripo, Meshy and Tencent Hunyuan3D have compressed the production cycle of a game 3D asset to "minutes per natural-language prompt". Higher throughput, however, has not brought controllable quality. AIGC outputs frequently contain non-manifold edges, zero-area faces and degenerate triangles that make up "triangle soup", as well as stretched or overlapping UVs and flipped normals, which trigger import warnings, failed lightmap baking, broken collision volumes and LOD chain failures in UE5 and Unity. The survey by Fukaya et al.[3] notes that the field still lacks a unified, production-ready quality assessment framework, and industry continues to rely on visual spot checks by artists, an approach that is subjective and cannot be embedded in a CI/CD-style content pipeline.

Three research lines have formed in the literature. The first builds datasets and dedicated models: AIGC-T23DAQA[4] covers six generators and 969 verified assets, collects human ratings along quality, realism and text-asset correspondence, and extracts shape, texture and text-alignment features from orbit-projection videos. The second treats a vision-language model as judge: Eval3D[5] characterises structural, semantic and geometric consistency through the disagreement between foundation-model representations such as DINO and Zero-123, while the cross-model VLM-judge protocol[6] cross-checks two model families and corrects for positional bias, achieving Cohen's kappa of 0.66; general-purpose vision-language assistants such as LLaVA[8] and Qwen-VL[9] have also been tried as zero-shot judges, with reported reliability far below that of task-specific protocols. The third line uses geometric and rendering metrics: Chamfer Distance, F-Score, CLIP Score[7] and FID are fully automatic, but they only reflect "resemblance" and cannot measure topological legality or UV bakeability, the engineering properties that actually decide whether an asset can enter production.

Three gaps remain. First, single-channel limitation: pure geometry cannot measure semantic alignment, and a pure VLM cannot detect non-manifold edges or UV defects, while dual-channel fusion remains unverified for game assets. Second, offline operation: quality tools have not been pushed down into the editor where assets are actually handled. Third, compute constraints: a university-level project cannot afford fine-tuning, so a lightweight path of "shallow models plus zero-shot calls to open multimodal models" is needed. This paper contributes: (1) a gradient defect injection protocol with deliberately overlapping severity intervals, reported under leakage-aware grouped repeated cross-validation and McNemar's test; (2) a 29-dimensional geometric descriptor set with an importance ranking; (3) a dual-channel architecture of "geometric hard rules plus multi-view semantics" that, calibrated under real CLIP weights, yields the two-sided finding that text-only zero-shot scoring is unusable whereas the embedding distance to reference renders works; (4) a UE5 plugin with end-to-end in-engine validation.

2. Method

2.1. Problem Definition and Overall Framework

Given a triangle mesh M=(V,F) with optional UVs and materials, the task outputs a triple (y,c,D), where y in {0,1} indicates usability, c in {A,B,C,D} the quality grade, and D the set of detected defects, each carrying a measured value, a threshold and a geometric location. Compared with a scalar score alone, the defect set makes the verdict traceable and directly actionable.

The overall framework is shown in Fig. 1 and comprises a representation layer, a decision layer and a delivery layer. The representation layer runs three channels in parallel: the geometric channel computes the 29 descriptors and feeds a baseline classifier; the multi-view semantic channel renders the mesh from several viewpoints and encodes the images with CLIP; the hard-

ISSN: 2688-8645

rule channel checks blocking defects one by one against an external threshold table. The decision layer fuses the scores and assigns a grade, with blocking defects holding veto power. The delivery layer writes the grade, score and defect list back into asset metadata and visualises them in an editor panel.

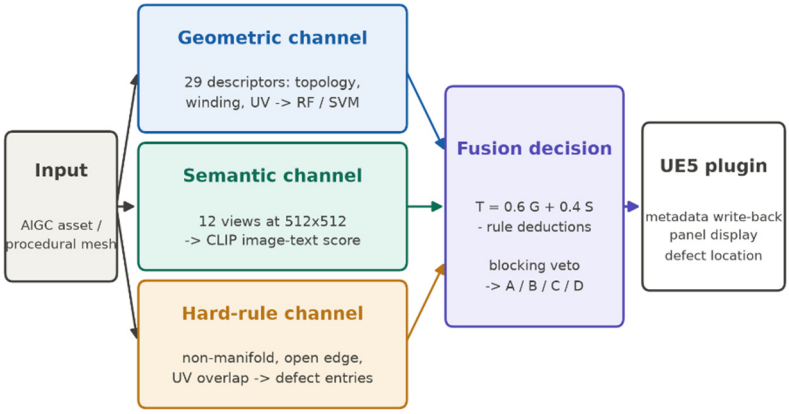


Fig 1. Overall framework of the dual-channel quality assessment

2.2. Gradient Defect Injection and Dataset Construction

Real AIGC assets are expensive to annotate and their defect distribution is skewed, so direct collection can hardly cover the "light defect" band that is hardest to judge. We therefore adopt a "clean prototype plus controlled corruption" strategy. Clean prototypes are six kinds of primitives from trimesh[10] (box, sphere, cylinder, capsule, cone, torus), five random instances each, with scale, aspect ratio and segmentation perturbed within reasonable ranges, giving 30 topologically valid assets with complete UVs; all are scaled to a unit bounding box to remove scale effects. Five defect types are injected into each prototype according to a continuous severity s, all growing linearly with s: Gaussian vertex noise with standard deviation $0.025 \cdot s \cdot L$ (L is the longest bounding-box edge); triangle winding flipped at ratio $0.20 \cdot s$; faces deleted to form holes at ratio $\max(0, s-0.35) \times 0.10$; triangles duplicated at ratio $0.03 \cdot s$; and UV overlap triggered with probability $\min(1, 1.2s)$ (all triangles sharing a single UV island).

The severity sampling intervals of the four grades deliberately overlap (Table 1). If the intervals were disjoint, the synthetic data would be almost linearly separable and a classifier could approach 100% accuracy, a result that means nothing for real scenes; overlapping instead simulates the ambiguous band between adjacent human grades, bringing the metrics closer to the noise level of real annotation.

Table 1. Injection strength of the five defect operators across four grades

Defect operator	Clean s in [0,0.22)	Light s in [0.12,0.42)	Medium s in [0.32,0.68)	Heavy s in [0.55,1.00]
Vertex Gaussian noise (sigma/L)	0-0.55%	0.30%-1.05%	0.80%-1.70%	1.38%-2.50%
Winding-flipped face ratio	0-4.4%	2.4%-8.4%	6.4%-13.6%	11.0%-20.0%
Face deletion (hole) ratio	0	0-0.7%	0-3.3%	2.0%-6.5%
Duplicated face ratio	0-0.7%	0.4%-1.3%	1.0%-2.0%	1.7%-3.0%
UV-overlap trigger probability	0-26.4%	14.4%-50.4%	38.4%-81.6%	66.0%-100%

ISSN: 2688-8645

Counting operators are rounded by round(n·ratio); when the count is below one, no defect is injected, so meshes with few faces in practice receive no deleted or duplicated faces at low grades. The dataset contains 120 samples, 30 per grade. Labels follow a five-point usability scale (clean 5, light 4, medium 2, heavy 1) binarised at 4, so light defects belong to the positive class and are the main source of confusion.

2.3. Geometric Descriptor Set

Descriptors are designed so that each one corresponds to a defect shape an artist can understand and repair. There are 29 dimensions in six groups (Table 2), and ratio-type descriptors are already dimensionless. Random-forest splits depend on value ordering and are insensitive to monotone transforms, so raw values are used directly; SVM-RBF requires standardisation first, otherwise scale dominates the kernel distance.

Table 2. The 29-dimensional geometric descriptor set

Group	Descriptors	Dim	Engineering meaning
Size and scale	n_vertices, n_triangles, surface_area, volume	4	Triangle budget, abnormal scaling
Bounding box	bbox_x, bbox_y, bbox_z, bbox_ratio	4	Flattened or stretched degenerate shapes
Triangle shape	tri_area_mean/std/min/max, thin_triangle_ratio, max_aspect_ratio, aspect_ratio_p95	7	Lighting and collision artefacts from sliver triangles
Topological validity	degenerate_face_ratio, duplicate_face_ratio, non_manifold_edges, open_edges, open_edge_ratio, n_components, largest_component_ratio, is_watertight, euler_number	9	Watertightness, importability, debris islands
Orientation and winding	winding_inconsistency_ratio (effective signal), inward_face_ratio (reference only)	2	Flipped faces, winding consistency
UV and material	has_uv, uv_overlap_ratio, uv_oob_ratio	3	Bakeability, colour bleeding, out-of-bounds UVs

The winding inconsistency ratio counts, over all shared internal edges, the proportion of edge-adjacent face pairs with opposite winding, and is an effective signal for flipped faces: a static-mesh importer reconstructs normals from winding, so inconsistent winding produces broken shading or even vanishing faces on one model, which in a rendered image often appears only as "odd lighting" and cannot be recognised reliably by eye. The inward-face ratio has more than 30% natural false positives on non-convex shapes such as tori and cones, so it is kept as a reference descriptor only and is excluded from rule decisions.

2.4. Multi-view Semantic Channel

Geometric descriptors characterise whether an asset can be used; the semantic channel answers whether it looks like what it should be. The mesh is rendered into multi-view images at equal horizontal intervals (the plugin defaults to 12 views at 512x512; this study uses 8 views at 256x256 to control batch rendering cost), and CLIP ViT-B/32[7] serves as a shared encoder with two switchable scoring forms. The first is text zero-shot: "finished" and "rejected" templates are constructed, a temperature softmax (temperature 10) is applied to the cosine similarities of the two templates, and the resulting probability of "more like a finished asset" is mapped to 0-100; views are aggregated as 0.7 x mean + 0.3 x worst view so that a single collapsed view is penalised even when the rest look normal. The second is reference-image distance: renders of qualified assets of the same category serve as references, the cosine

ISSN: 2688-8645

similarity between image embeddings is computed view by view and averaged to give d=1−cos, which is then linearly mapped to a 0-100 score. The first form needs no asset library; the second depends on a trustworthy library of qualified assets. Measurements in Section 3.5 show a large gap in discriminative power between them, and the second is recommended in practice. A further value of this channel is coverage of semantic misalignment that geometric descriptors cannot express (for example, a chair requested but a table produced): topologically valid but semantically failed.

2.5. Fusion Decision and Grading

The total score is composed by Eqs. (1) and (2), where R is the rule score, with blocking, major and hint defects deducting 40, 15 and 5 points respectively and 0 as the lower bound; P is the classifier probability of "usable" times 100; w_r=w_b=0.5; S is the semantic score; and the defaults are α=0.6 and β=0.4, adjustable by configuration. Defect deductions enter the total only through R and are not subtracted again in Eq. (1). If any blocking defect is present, the asset is graded D directly; otherwise a total of at least 85 is grade A, at least 70 grade B, at least 55 grade C, and anything lower grade D. When the semantic channel is unavailable, α and β are renormalised to 1 and 0, degrading to pure geometric assessment so that the pipeline never breaks in any environment. Blocking thresholds (non-manifold edges greater than 0, missing UVs, triangle budget exceeded, bounding-box aspect ratio above 30, and so on) are all externalised into a JSON configuration, so the art team can adapt them to project conventions without touching code.

$$T = \alpha(w_r R + w_b P) + \beta S \quad (1)$$

$$R = \max(0, 100 - \Sigma p_d) \quad (2)$$

2.6. UE5 In-Engine Plugin Implementation

The plugin is built on the UE5 Python Editor Script, with the chain: query static meshes through AssetRegistry, export OBJ in batch, invoke an external Python interpreter in a subprocess that reuses exactly the same feature extraction and inference code as the offline experiments, write scores back as Metadata Tags and read them back one by one for verification, and finally display results in a panel with navigation to the Content Browser. The key design is "offline and online share one source", which avoids a mismatch between "the model in the paper" and "the model in the plugin"; subprocess isolation keeps inference from blocking the editor main thread.

3. Experiments

3.1. Experimental Setup

The dataset is the 120 samples built in Section 2.2, with 29 descriptors and a positive rate of 0.50. Samples are four variants derived from each of 30 prototypes, and variants of the same prototype share base topology and scale; if folds were split randomly by sample, the same prototype would appear in both training and test sets and the model could "memorise the shape" rather than "learn to detect defects". Two protocols are therefore compared: (a) StratifiedKFold, split by sample; (b) StratifiedGroupKFold, grouped by prototype. Each is repeated 10 times (seeds 2026-2035), and mean plus minus standard deviation is reported, with McNemar's exact test on the out-of-fold predictions of a single split. Three methods are compared: rule-only verdicts (only the threshold rules of Section 2.5, no learning component), SVM-RBF (StandardScaler plus RBF kernel, C=4.0, balanced class weights), and Random Forest (300 trees, min_samples_leaf=2). The environment is an ordinary desktop workstation, with open-source trimesh[10] and scikit-learn [11]; semantic experiments run on CPU with the official clip-vit-base-patch32 model (151.3 M parameters).

ISSN: 2688-8645

3.2. Comparison with Baseline Methods

Table 3 and Fig. 2(a)(b) give results under the grouped protocol: Random Forest reaches 88.33% accuracy and 0.8814 F1, SVM-RBF 85.83% and 0.8682, and rule-only verdicts only 66.67% and 0.5918. Over 10 repetitions, Random Forest achieves 88.00%±0.70% (F1=0.8792±0.0071) under the grouped protocol and 85.33%±0.70% (F1=0.8523±0.0075) under the stratified one.

Table 3. Performance of three methods on the usability binary task (grouped five-fold, out-of-fold predictions)

Method	Accuracy	Precision	Recall	F1
Rule-only	0.6667	0.7632	0.4833	0.5918
SVM-RBF	0.8583	0.8116	0.9333	0.8682
Random forest	0.8833	0.8966	0.8667	0.8814

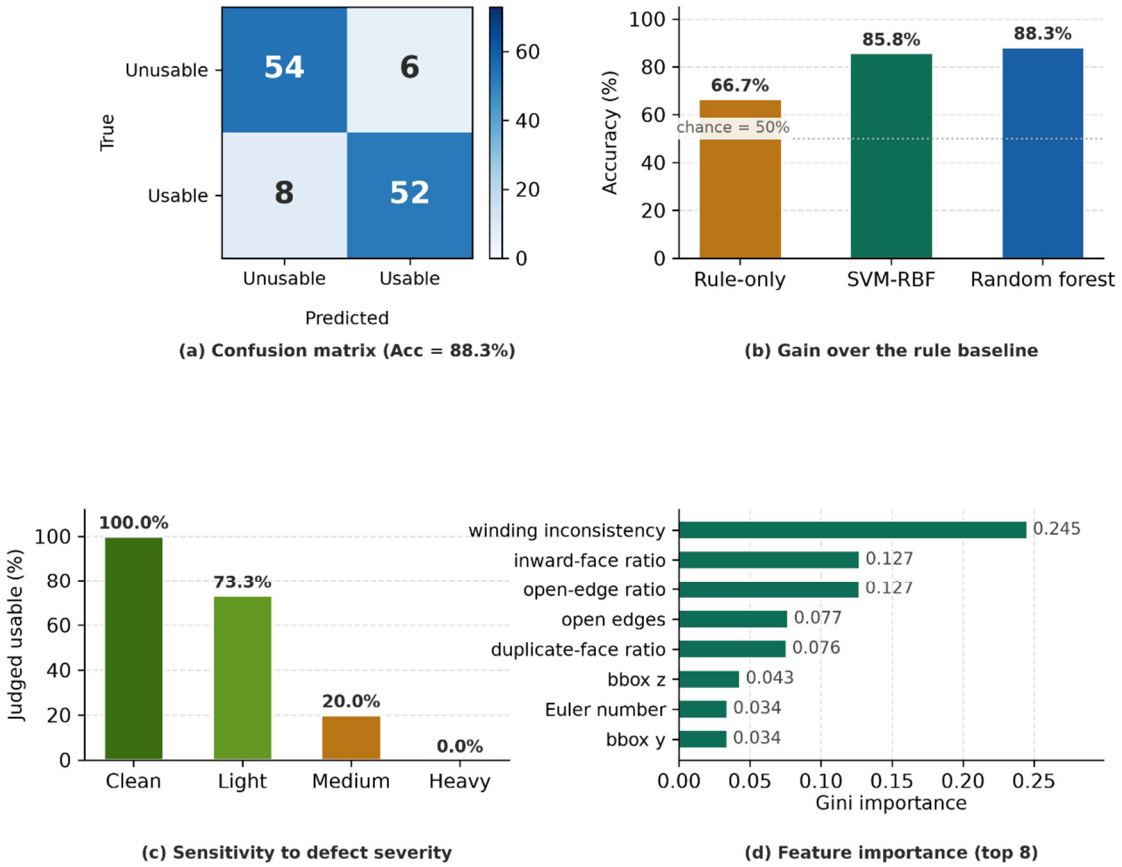


Fig 2. Experimental results on the validation set

The grouped protocol is not lower than the stratified one; it is 2.67 percentage points higher, contrary to the intuition that splitting by sample must inflate scores through leakage. Defect injection is itself random, the severity intervals of the four variants of a prototype overlap (Table 1), and identical topology does not uniquely determine the label. The conclusion is therefore that the leakage risk exists structurally but does not translate into metric bias on this data; nor can one infer from "grouping did not lower the score" that leakage is absent. McNemar's test on the out-of-fold predictions of a single split shows Random Forest uniquely correct on 8 cases and SVM-RBF on 5, with p=0.581, so this paper does not claim Random Forest to be superior to SVM-RBF. Against the rule-only baseline, Random Forest improves accuracy by 21.33 percentage points and F1 by 0.290. On the confusion matrix, Random Forest is

balanced with 6 false positives and 8 false negatives; SVM-RBF has 13 false positives and a recall of 0.9333, tending to accept light defects and thus carrying higher rework risk in production; the rule baseline has a recall of only 0.4833, because fixed thresholds cannot adapt to primitive shape—the same proportion of open edges is normal structure on a torus but a hole on a box.

3.3. Sensitivity to Defect Severity

Table 4 and Fig. 2(c) show that the proportion judged usable by Random Forest decreases monotonically as defects worsen: 100% for clean, 73.3% for light, 20.0% for medium and 0% for heavy. The 20.0% at the medium grade shows that the model does not make a binary judgement on "whether defects exist" but responds continuously to accumulated defect severity. The 73.3% at the light grade has an engineering meaning: repairing a light defect usually costs less than regenerating the asset, so judging it usable is reasonable; the remaining 26.7% have already triggered blocking rules or fall into the medium distribution, consistent with the prior that the light grade is the ambiguous annotation band.

Table 4. Proportion judged usable at each defect grade (grouped five-fold, out-of-fold)

Defect grade	Samples	Judged usable	Usable rate
Clean	30	30	100.0%
Light	30	22	73.3%
Medium	30	6	20.0%
Heavy	30	0	0.0%

3.4. Feature Importance

Table 5 and Fig. 2(d) give Gini importance: winding inconsistency ratio 0.2452, inward-face ratio 0.1273, open-edge ratio 0.1271, open-edge count 0.0769 and duplicate-face ratio 0.0757, with the top five accumulating to 0.6522; no size descriptor enters the top ten, showing that "how large the model is" matters less than "whether the model is self-consistent". Two qualifications are needed. First, Gini importance favours continuous descriptors with many split points and spreads weight across collinear descriptors, so Table 5 reflects only relative contribution on this dataset. Second, the top rank is directly tied to the injection protocol: winding flipping is the only operator that stays continuously non-zero across all grades, so one cannot conclude from it that quality control should look at winding first (that would be circular); the inward-face ratio ranking second likewise stems mainly from natural concavities of non-convex shapes.

Table 5. Random-forest feature importance (top 5)

Rank	Descriptor	Gini importance
1	winding_inconsistency_ratio	0.2452
2	inward_face_ratio	0.1273
3	open_edge_ratio	0.1271
4	open_edges	0.0769
5	duplicate_face_ratio	0.0757

3.5. Calibration and Fusion Ablation of the Semantic Channel

To answer whether the semantic channel truly adds value, three groups of experiments are run under real weights.

(i) Latency. The cost of official CLIP weights on CPU is given in Table 6: per-image encoding takes 90.48/93.38/77.55/73.31 ms at batch sizes 1/2/4/8, an 8-view asset costs about 0.76 s (757 ms measured including decoding), and model loading takes 0.41 s. Although this is roughly

ISSN: 2688-8645

two orders of magnitude higher than the geometric channel, 200 assets take about 2.5 min in a single process and can be scheduled as overnight batch jobs.

Table 6. Local inference cost of CLIP ViT-B/32 with official weights

Item	Configuration / result
Model	ViT-B/32 (official pretrained weights)
Parameters	151.3 M / 605 MB
Input resolution	224x224 (test image 512x512)
Device	CPU
Encoding latency per image	90.48 / 93.38 / 77.55 / 73.31 ms (batch 1/2/4/8)
Per asset (8 views, incl. decoding)	757 ms

(ii) Two scoring forms. Text zero-shot reaches an AUC of only 0.2725, and its semantic score rises monotonically as defects worsen (41.6 to 42.7 to 43.4 to 44.3), meaning the model systematically judges assets with more broken faces and spikes as more "like a finished product". This agrees with the rendering-CLIP discriminability near chance (0.48) reported in [6]: CLIP's image-text alignment is dominated by the overall style of "grey 3D renders", and geometric damage instead raises the match with "rich in detail", so defect semantics never enter the text direction. Reference-image distance behaves completely differently: AUC 0.8994, 84.17% accuracy at d of at most 0.0687, with the four grade means 0.0177/0.0519/0.0834/0.1221 strictly monotone, already close to the 29 geometric descriptors (AUC 0.9193).

Table 7. Measured comparison of the two semantic scoring forms (120 samples, grouped out-of-fold)

Form	Mean by grade	AUC	Accuracy
Text zero-shot (dual prompts)	S 41.6 / 42.7 / 43.4 / 44.3 (reversed)	0.2725	0.5000
Reference-image distance (same prototype)	d 0.0177 / 0.0519 / 0.0834 / 0.1221 (monotone)	0.8994	0.8417

This shows that when a multimodal model is used for quality control, "asking the model to judge good from bad in natural language" is unreliable, whereas "using the model as an encoder and comparing against qualified assets already in the library" is far more robust. One caveat: the reference experiment uses clean meshes of the same prototype, the most favourable setting for this channel (an upper bound); real deployment must retrieve references by category, and retrieval error will lower the metrics.

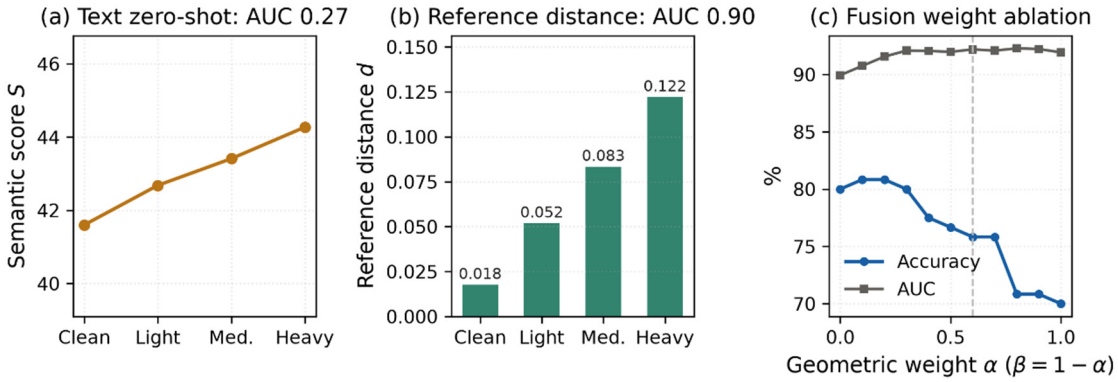


Fig 3. Calibration of the semantic channel and ablation of the fusion weight (dashed line in (c): plugin default α=0.6)

ISSN: 2688-8645

(iii) Fusion ablation. Scanning α in steps of 0.1 in Eq. (1) with S=100(1−d/0.20) gives Table 8 and Fig. 3(c). AUC peaks at α=0.8 (0.9228), only 0.0035 above the single geometric channel (0.9193); under the plugin default rule "total score at least 70", α=0.2 gives 80.83% accuracy, the geometric channel alone 70.00% and the semantic channel alone 80.00%. The two curves peak at different points, showing that the channels have comparable and highly correlated discriminative power: geometric descriptors measure topology and winding directly, while reference distance measures the projection of the same defects onto rendered images, so the two are not complementary. The semantic channel therefore cannot replace the geometric one; its value lies in reaching comparable discriminative power with very little feature engineering, making it suitable as a redundant check and fallback.

Table 8. Fusion weight ablation (T=α·G+β·S, grouped out-of-fold)

α (geometric)	β (semantic)	Accuracy (T of at least 70)	AUC
0.0	1.0	0.8000	0.8994
0.2	0.8	0.8083	0.9156
0.4	0.6	0.7750	0.9206
0.6	0.4	0.7583	0.9219
0.8	0.2	0.7083	0.9228
1.0	0.0	0.7000	0.9193

One more point must be made: under the fixed threshold "total score at least 70", the geometric channel reaches only 0.7000 accuracy, clearly below the 0.8833 that the same classifier achieves at a decision threshold of p=0.5 (Table 3). The difference comes from an uncalibrated score-to-grade mapping: G=0.5R+0.5P gives clean samples a mean total of only 80.3, placing them in grade B rather than A. Before formal deployment, grade cut-points should be recalibrated on validation-set quantiles, or the classifier probability should be used for the decision directly with the 0-100 scale kept for display only.

3.6. End-to-End In-Engine Validation

The plugin chain is validated in a UE5 5.4.4 teaching project (Table 9): among 1873 assets in the Content Browser, the plugin identifies and loads 223 static meshes; 8 samples are taken through the full chain, with OBJ export 8/8, external process return code 0, and write-back plus read-back verification 8/8, over 6.52 s end to end, under 1 s per asset on average, the bottleneck being OBJ serialisation and subprocess communication.

Table 9. Measured results of the in-engine quality-audit pipeline in UE5

Stage	Result
Engine version	UE5 5.4.4
Total assets in content browser	1873
Static meshes identified / loaded	223 / 223
OBJ export success rate	8 / 8
Return code of external inference process	0
Score records parsed	8
Metadata write-back and read-back verification	8 / 8
End-to-end latency	6.52 s
Grade distribution	6 grade A, 2 grade B
Mean / min / max audit score	95.0 / 80.0 / 100.0

A batch scan yields a further 23 records with a mean score of 94.78, 17 grade A and 6 grade B; these are not 23 independent assets but the base, _Internal and _UV1 variants of the same 8 source assets. This validation round is completed with the semantic channel disabled (the

ISSN: 2688-8645

channel field of every record is geometry), so the conclusion is limited to the geometric channel and pipeline connectivity; all of these records are judged usable and none falls into grade C or D, which shows that the thresholds face no rejection pressure on teaching assets that are already qualified, and that the plugin's ability to reject unqualified assets has not yet been examined in the engine.

4. Summary

4.1. Why the Rule Baseline Fails

The rule-only baseline reaches only 66.67%, often misread as "the rules are poorly written". The real reason is that thresholds are global, whereas defect acceptability depends heavily on asset category and purpose: terrain chunks legitimately allow open boundaries, single-sided foliage legitimately allows non-watertight geometry, and the same descriptor on a character prop is a hard defect. The classifier's gain is essentially learning the conditional relation "descriptor combination, category, acceptability", which no one-dimensional threshold can express. The correct role of the rule channel is therefore not "primary scorer" but "interpretable fallback and locator".

4.2. Comparison with Existing Benchmarks

Compared with AIGC-T23DAQA[4], this work does not target correlation with human subjective ratings but the decision "can it enter the production pipeline", and therefore introduces engineering properties such as non-manifold edges and UV bakeability. Compared with Eval3D[5], it does not pursue pixel-level interpretability but pushes interpretability down to defect entries and geometric locations, closer to the repair actions an artist takes. Compared with the cross-model VLM-judge protocol[6], it trades very low compute cost for engineering usability, at the price of coarser semantic judgement. The three correspond respectively to "score accurately", "score finely" and "score fast and land it in production". In addition, [6] reports that rendering-CLIP proxies reach only 0.48 accuracy on pairwise preference tasks (equivalent to chance); our measurement echoes this: text zero-shot AUC is only 0.27, and introducing qualified references raises it to 0.899—same model, same rendered images, the only difference being whether a reference is introduced—indicating that the usability of this route depends on constructing the reference system rather than on the capability ceiling of the model itself.

4.3. Limitations and Future Work

First, external validity is untested: all samples come from procedural primitives with injected defects, and real outputs of Tripo, Meshy and Hunyuan3D have not entered the experiments. Second, construct validity suffers from a shared source: labels are determined by the injected severity s, and the 29 descriptors are generated from s through the same operators, so the classifier largely learns "infer s from descriptors"; overlapping intervals only mitigate this and cannot remove it, and the accuracy reported here should be read as "the ability to distinguish injected defect severity under the protocol of this paper". Third, the reference-image experiment of the semantic channel is an upper-bound setting, and both the grade mapping and the default α are uncalibrated (see Section 3.5). Fourth, with only 120 samples against 29 descriptors and topology shared across variants of a prototype, the stability of the importance ranking needs rechecking. Fifth, no agreement comparison with artist ratings (such as Cohen's kappa) has been made. Future work will collect real generator outputs to build a held-out set, calibrate the channel weights, and introduce parameter-efficient fine-tuning[12] to improve the adaptability of the semantic channel.

ISSN: 2688-8645

5. Conclusion

Targeting the production-readiness decision for AIGC 3D game assets, this paper builds a validation set by gradient defect injection and a 29-dimensional geometric descriptor set, compares the rule, SVM-RBF and Random Forest paths, calibrates the semantic channel under real CLIP weights, and implements a UE5 in-engine plugin. Under repeated five-fold cross-validation grouped by prototype, Random Forest reaches 88.00%±0.70% accuracy and 0.8792±0.0071 F1, 21.33 percentage points above the rule-only baseline; the grouped protocol is 2.67 percentage points higher than the stratified one, so no leakage-driven inflation is observed, and the difference from SVM-RBF is not significant (p=0.581). Usable rates across the four grades, 100%, 73.3%, 20.0% and 0%, respond monotonically. The semantic channel is two-sided: text zero-shot scoring reaches an AUC of only 0.27 and in the wrong direction, whereas the embedding distance to qualified reference images reaches AUC 0.899 and 84.2% accuracy; fusion brings an AUC gain of only 0.0035, so the two channels are largely redundant. In the engine, 8 assets pass the full chain 8/8 in 6.52 s. The value of this work lies not in setting a new record on metrics but in providing a reproducible, interpretable and deployable quality assessment path under a low compute budget, and in clarifying with measured data the applicable boundary of "using a multimodal model to judge quality directly".

References

[1] Poole, B., Jain, A., Barron, J. T., & Mildenhall, B. (2023). DreamFusion: Text-to-3D using 2D diffusion. In Proceedings of the International Conference on Learning Representations.

[2] Jun, H., & Nichol, A. (2023). Shap-E: Generating conditional 3D implicit functions. arXiv:2305.02463.

[3] Fukaya, K., Daylamani-Zad, D., & Agius, H. (2024). Evaluation metrics for intelligent generation of graphical game assets: A systematic survey-based framework. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(12), 7998–8017. https://doi.org/10.1109/TPAMI.2024.3400447.

[4] Fu, K., Duan, H., Zhang, Z., et al. (2025). Multi-dimensional quality assessment for text-to-3D assets: Dataset and model. IEEE Transactions on Multimedia, 27, 8145–8159. https://doi.org/10.1109/TMM.2025.3478612.

[5] Duggal, S., Hu, Y., Michel, O., et al. (2025). Eval3D: Interpretable and fine-grained evaluation for 3D generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13326–13336).

[6] Asaria, A., Salomone, T., & Gandhi, D. (2026). A cross-model VLM-judge protocol for single-image 3D mesh quality. arXiv:2606.18451.

[7] Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. In Proceedings of the 38th International Conference on Machine Learning (pp. 8748–8763).

[8] Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual instruction tuning. In Advances in Neural Information Processing Systems.

[9] Bai, J., Bai, S., Yang, S., et al. (2023). Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv:2308.14466.

[10] Dawson-Haggerty, E., et al. (2019). trimesh: Python library for triangular meshes. https://github.com/mikedh/trimesh.

[11] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://doi.org/10.5555/1953048.2078195.

[12] Hu, E. J., Shen, Y., Wallis, P., et al. (2022). LoRA: Low-rank adaptation of large language models. In Proceedings of the International Conference on Learning Representations.